Development of a Navigation System for an Autonomous Guided Vehicle using Android Technology

C J Snyman, TI van Niekerk
Department of Mechatronics
Nelson Mandela Metropolitan University
South Africa

Abstract—Modern cell phone hardware, due to its integrated peripherals, provides a low cost intelligent controller for use in the navigation of an Automated Guided Vehicle. Most commercial AGV's use proprietary hardware which is expensive to replace and also difficult to maintain. Using industrial hardware components combined with Android mobile platforms could provide a low-cost alternative. This would be easier to maintain, using existing in-house factory maintenance knowledge. A prototype AGV was designed and developed based on an integrated system between an industrial Programmable Logic Controller and an Android operating system mobile platform. This system utilises the mobile platforms integrated Global Position System or video camera as tools for navigation. Experimental tests were performed to determine whether the prototype can navigate a predefined course by making use of GPS and camera line following algorithms. The test results were very positive, but highlighted the limitations in the accuracy of cell phone based GPS receivers.

Index Terms— Automated Guided Vehicle, Android, Global Position System, Programmable Logic Controller, Navigation

1 INTRODUCTION

AUTOMATED Guided Vehicles (AGV)'s have been in existence since the 1950's, where when analogue technologies were initially used to create an unmanned mobile machine that was able to avoid collisions [1]. The purpose of these AGV's was to move without a human driver and to perform tasks automatically. Initial projects were aimed at military use, but soon expanded to the industrial sector.

The use of mobile AGV's has increased considerably over the last decade. This is mainly due to the flexibility they provide in the increasing demands of the factory environment. The use of an AGV can be seen as a mobile conveyer belt system for material handling. Owing to its immediate benefit within the factory environment, research in the AGV and its associated navigation techniques has been focused on indoor use.

Unfortunately most of the commercially available AGV's have critical shortcomings. These include issues such as closed architecture, incomplete documentation and difficulty in adapting to increased demands [2].

Android is an operating system which is based on the proven Linux kernel. It was primarily designed to be used on mobile devices (smartphones and tablet computers), but has since been implemented on various embedded systems that range from televisions and wristwatches to household appliances.

This paper suggest the use of Android mobile device to control an AGV because of the wide variety of sensors available (compass, proximity sensor, ambient light sensor, camera, Bluetooth, GPS,Wi-Fi, USB). The review investigates whether mobile devices will meet the requirements of what Pancham [3] describes as an ideal AGV, but owing to its relatively low cost and size, the advanced processing power of modern cell phones with integrated GPS should provide the ideal intelligent controller for use in an AGV.

The overall aim of this project is to integrate the Android operating system mobile hardware technology and software features with industrial standard drive and control options with the view to creating an autonomously driven AGV.

2 SYSTEM ARCHTECTURE

The final research AGV architecture implements the PLC for all the motor control I/O. The Android mobile platform will do all the calculation required for the navigation algorithm by making use of its integrated sensors (GPS and Compass) to accomplish this task. The Figure 1 block diagram illustrates the relationship between the PLC and

the Android mobile platform, highlighting their individual responsibilities.

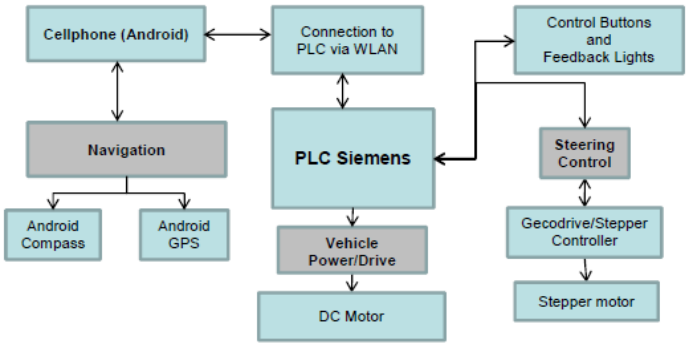

Figure 1: Block Diagram of System Architecture

2.1 Mechanical Components

The use of a three wheel platform was selected to replicate the steering used in material handling vehicles, with the front wheel providing the steering. This decision would provide opportunities to convert existing material handling vehicles into AGV's, using similar techniques to those used in the research AGV. The frame of the prototype vehicle was built out of aluminium extrusion, which is simple to assemble and still provides a strong frame that could handle collisions during the testing phase.

The front wheel castor is held in position using pillow block bearings. A stepper motor is connected to the front wheel, using a drive belt, with a pulley ratio setup for increased torque (using a smaller pulley on the motor and a bigger pulley on the castor). The higher torque ratio was selected to ensure that the stepper motor will be able to turn the front wheels on various friction surfaces.

A solid drive shaft is used to connect the driving wheels. To assist in cornering, owing to not having a differential, one of the drive wheels has been left unfastened. This is to allow for slippage when cornering. The drive shaft is connected to the drive motor using a belt with a one-to-one pulley ratio.

2.2 Electrical System

The research AGV hardware controller was selected on the criterion that it should be easily maintainable and upgradeable. A decision was made to use an industrial Siemens PLC (S7-300) for use as the hardware controller, owing to its wide use in the industry and the ease with which it could be upgraded. The S7-300 PLC was selected for its integrated PWM motor control and also had enough IO for the other peripherals.

A Gecko drive G203V controller was selected owing to its economical price and simple installation. The PLC pulses the stepper controller at high speed using PWM to provide steering control. The number of PWM pulses is recorded using a high speed hardware counter. The stepping direction is controlled by a hardware digital output to the stepper controller.

The design of the drive motor control circuit is based on the use of an H-bridge circuit, but implemented using a DPDT relay. The motor direction is controlled by 24V signal from the PLC. A transistor is used to switch the 12V motor supply with a 24V PWM signal. The transistor is used to allow fast switching due to the PWM signal.

A feedback panel contains one selector switch and five push buttons, four of the buttons having built-in lights which are all wired to the PLC I/O. The purpose of the feedback panel is to assist during the development stage and to help diagnose problems in the software. The red button is used as the E-Stop switch, with the red light flashing during alarm conditions. The selector switch is used to switch between manual control of the research AGV or automated guidance. The black button is used to clear any alarm conditions.

The research AGV makes use of a normal LAN setup, combining WLAN for integration to the Android phone. This setup would also allow for future integration of a SCADA system.

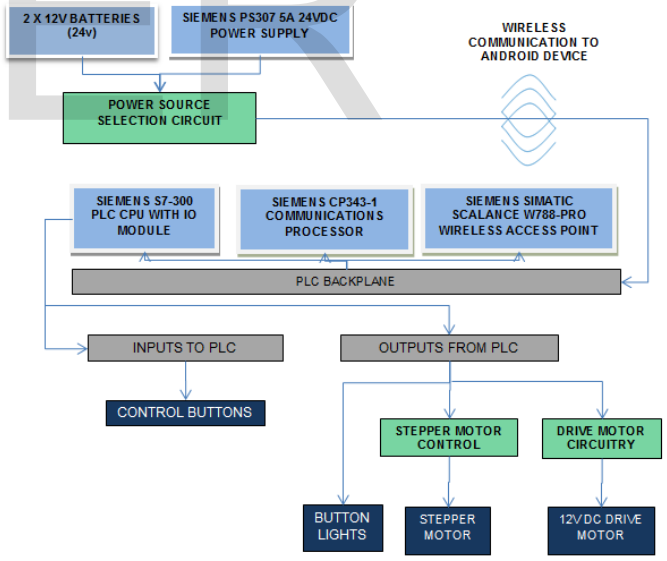

Figure 2: Block Diagram of Research AGV Electrical System

3 SOFTWARE ARCHITECTURE

The research AGV software is divided into the Android and he PLC sections. The Siemens PLC software is responsible or the motor control and safety. The PLC listens for messages sent from the Android mobile platform and then executes them accordingly. The Android mobile

platform takes care of the navigation algorithm and will then send the appropriate steering command to the PLC. Figure 3 is a block diagram illustrating the individual platform responsibilities as well as how the two software platforms communicate in order for the AGV to function.

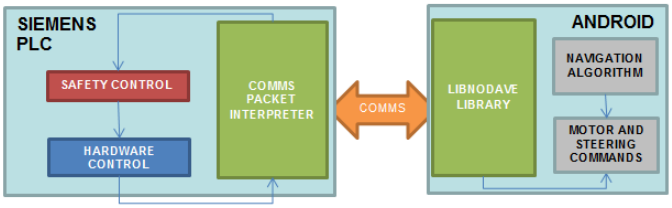

Figure 3: Block Diagram Software Architecture

3.1 PLC Software Operation

The software on the PLC listens for commands sent from the Android mobile platform after start-up. The software will only control the steering and drive motors if a command is received or if a safety condition has been breached. Figure 4 indicates the flow of decisions during the PLC software process. Once a communication packet has been received, the appropriate steering limits will be set. Before any of the motors are switched on, the software will check to see whether all the safety conditions have been met. The motors will stay on until the steering limit, sent by the command, is reached. During the 'motor on' condition, the safety conditions are continuously checked and will remove power to the motors if any problem is sensed.

Knowing the current position or direction of the steering wheel is very important for automated navigation. The research AGV steering hardware uses a stepper motor requiring pulse steps in order for it to move. The software counts the steps (pulses) sent to the stepper controller using the Siemens Special Function Block 47 (SFB) for high speed counting and keeps track of the direction of those steps/pulses. The software allows for four different step positions on the left as well as the right of the vehicle. When the research AGV starts-up, the front wheel is centred, giving it a known starting position in the centre and a reference pulse value. Depending on the steering direction command send, the stepper motor is pulsed in the appropriate direction until it reaches the steering limit value that was predefined. Using this technique, it is very easy to determine the steering wheel direction by looking at the pulse count value of the stepper motor. The steering control software is designed to allow for direction changes or commands while the wheel is moving towards a limit (in other words before the command has been completed), allowing for an immediate direction change if it is required.

The software to control the drive motor is simplistic. The software implements the Siemens SFB49 (Special Function Block) which utilises PWM to control the speed of the drive motor. The direction of the motor is controlled with a normal digital output that switches the H-bridge circuit.

The software implements alarm control by using an alarm flag. If the alarm flag is triggered, all motors are stopped immediately and the red E-STOP light will flash. During the alarm state condition, all the communication protocol commands will be ignored until the alarm state has been acknowledged using the black button on the console. Once the alarm condition has been cleared, the prototype will continue to listen for commands from the communication protocol.

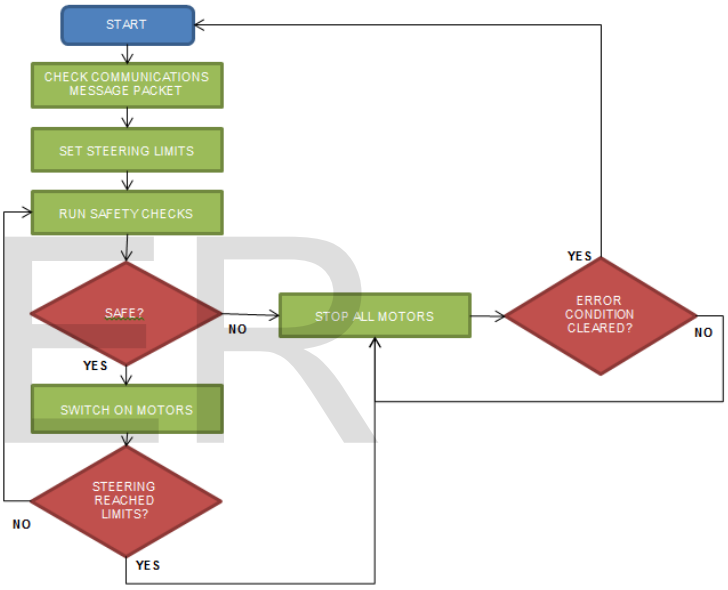

Figure 4: Flow Chart of PLC Software Operation

3.2 Android PLC Communication

The communication between the PLC and Android is done over a wireless IP (Internet Protocol) connection. The PLC and Android device are setup on the same network and therefore can communicate as if there were a physical connection.

The communication to control the PLC from the Android device can be compared to that of a SCADA (Supervisor Control and Data Acquisition) system. The communication between the Siemens PLC and the Android device is handled using the open source Java Libnodave library, written by Thomas Hergenhahn. The communication library is used to facilitate data exchange between the Android device and Siemens PLC. The library contains all the required functions to connect to the PLC and to read or

write to specific memory areas of the PLC. The library is used to change PLC address values directly and therefore forms the basis of the communication protocol.

Using the Libnodave library, the Android application can manipulate address values directly. The memory word addresses M125 and M126 inside the PLC have been selected for use as command locations. Twelve commands have been identified for use between the Android device and the PLC. The commands are used to control the motors of the research AGV, utilising four different left position commands and four different right position commands. The command set also includes a centre steering command and manual commands for left and right turning. The software on the PLC scans these memory locations continuously to see whether any of the commands are active using a high speed cyclic interrupt. Any active commands are executed and on completion the bit will be cleared waiting for the next command.

3.3 Android Software Operation

Once the Android application is installed, it can be selected under the Applications Window on the mobile device. The Main Menu consists of three buttons, each button linking to the appropriate navigation technique or the manual control selection.

The manual control section of the Android application is used during diagnostics and allows the operator to control the research AGV remotely. This section of the application communicates to the PLC using only three commands. The two manual steering commands allow full control of research AGV steering wheel and are not limited to predefined steering limits.

The GPS control section of the application is called when the GPS control button, in the Main Menu, is pressed. While the GPS control activity starts, the application checks to see whether the GPS of the device has been enabled. If the application detects that the GPS has not been enabled, it will open the Settings Menu to assist the user in enabling it. The GPS control displays a Google map of the research AGV's current location. The current location GPS coordinates are displayed at the top of screen, with the distance to the next waypoint being displayed at the bottom of the screen. A start button is located on the screen, which initiates the automatic guidance to the next waypoint. Once this button is pressed, it will change to a stop button, which will allow the user to stop the research AGV manually. This button will change back to a start button if either the stop button has been pressed or if the waypoint has been reached. The menu button on the mobile contains a button to mark a new waypoint. If pressed, the current position of the Android mobile will be marked as the new waypoint.

The GPS control activity functions by using four different threads which all run concurrently. The four threads are GPS-, compass sensor-, display- and navigation updates. Different threads are used to ensure that sensor updates are independent and to prevent delays due to the calculations required for the navigation algorithm. The GPS and compass updates monitor the sensors for any location changes and store the latest available location information. The display thread will use the latest location information stored to update and give feedback to the user. The navigation updates do all the GPS navigation algorithm calculations and then send the appropriate commands to the PLC, in order to steer the research AGV.

The line follower section of the application is called when the Line Follower button is pressed on the Main Menu. The application was developed to detect a black line on a light surface and then to steer according to that detected line. The use of the Android platform required an alternative approach to traditional line following methods. The line following algorithm makes use of the Android camera to detect the line that needs to be followed, using the technique shown by Kelley [4]. The application uses the flash LED (Light Emitting Diode) from the mobile platform to eliminate any external light sources. A black and white camera view is displayed on screen once the program starts, with a yellow line displaying the detected line.

4 NAVIGATION

The research AGV implements two different navigation techniques, namely a GPS based algorithm and a line following algorithm.

4.1 Android Line Following Algorithm

Unlike traditional line following techniques, the implemented line following algorithm makes use of the Android camera to detect the line. The camera view is displayed on the Android application view and is then analysed to locate a black line. Analysis of the camera view involves the understanding that LCD (Liquid Crystal Display) pixels use a combination of the RGB (Red Green Blue) colour pixels. The brightness of each of the three colours is controlled to produce the different colours. Each of the colour's brightness can be detected in software and will be used to locate the black line. In order to display black, the brightness of all three of the colours is at its lowest.

The first step of the software algorithm is to convert the normal colour image to greyscale. The camera image is then rotated 90 degrees in order for the view to display upright

on the Android application, since the software returns it with a 90 degree offset. The algorithm then determines what position represents the top third of the screen.. This position is used to scan the width of the screen to detect the black line. The complete screen width at the height selected, will be scanned. Each pixel's red colour component brightness value is checked to find the lowest value. The lowest pixel value position is stored. The complete screen is not scanned to minimise calculation time. The darkest position found is compared to the middle of the screen. This will then indicate in which direction the research AGV needs to steer. The appropriate steering command turning limit is selected, based on the distance between the middle of the screen and the position detected.

4.2 Android GPS Navigation Algorithm

This algorithm uses both the GPS sensor and the compass sensor that is integrated into the Android mobile platform. Both sensors are continuously monitored in software to ensure that the most recent location information is used inside the algorithm. The algorithm used, which is adapted from Barrette [5], is designed to navigate the research AGV to within three metres of its destination.

The algorithm process, as illustrated in Figure 5, starts by loading waypoint information. The latest compass heading and GPS location information are then retrieved from the research AGV. Using this information, the heading as well as the distance to the next waypoint is calculated. If the distance to the next waypoint is more than three metres, the correct steering commands will be sent to navigate the research AGV closer to its destination. If the distance between the research AGV and the waypoint is closer than three metres, the software will send a command to stop all motors or load the next waypoint if there is more than one allocated.

5 TESTING AND VERIFICATION

5.1 Line Following Algorithm Test

The line following algorithm calculates where the navigation black line is located on the camera. The algorithm then sends the appropriate steering command to follow the line. While the algorithm is running, the mobile device LED is switched on to eliminate external light conditions. The purpose of these tests is to determine whether the research AGV can follow a black line on a light surface.

Initial tests revealed that the angle of the Android device with the ground is very important. Mounting the Android device parallel to the floor caused problems. The reflection of the LED at this angle blinds the camera view and it is unable to detect the black line consistently. Mounting the Android device at an approximate 45 degree angle created the best results.

Initial test also revealed that he steering control was too slow, and the research AGV was moving too fast. This resulted in the research AGV in overshooting bigger bends in the guide path. Due to the problems the research AGV steering control was redesigned to move at much a higher speed to try and improve the test results.

Another possible problem was the small view area of the camera. If the view area is too small, the guide line can be lost with a very small turning angle. To try and eliminate this, the Android algorithm was changed to operate using the landscape mode of the Android device. The Android device was also mounted much higher from the ground to increase its viewing area.

The final test results were much improved. The algorithm in its final state was able to consistently follow the test guide path. However this was only achievable at low speed and with a very simple test guide path.

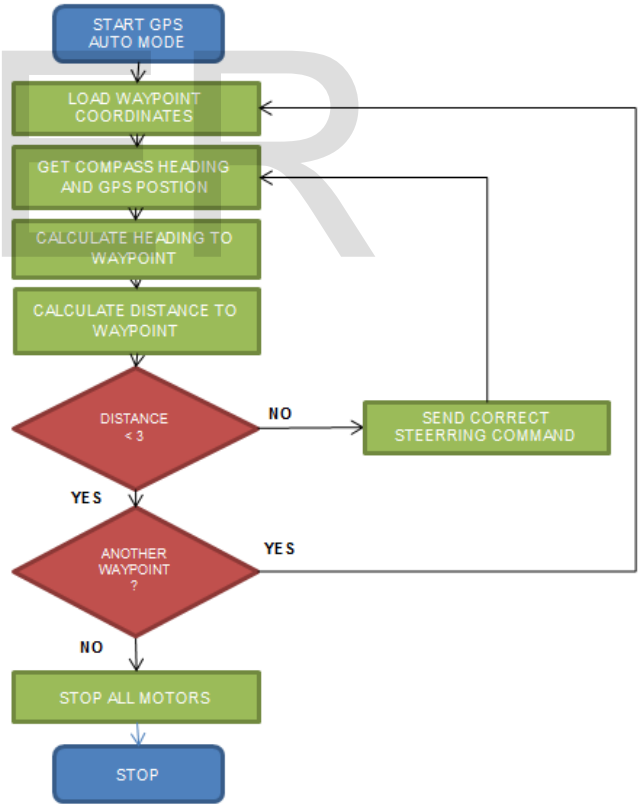

Figure 5: GPS Navigation Flow Chart

5.2 Android GPS Navigation Algorithm Test

The GPS navigation algorithm, implements the Android device GPS and electronic compass to navigate the research AGV to a predefined GPS location. The purpose of the

following tests is to determine whether the algorithm can successfully and accurately navigate to predefined waypoints.

The first test was done to determine whether the calculations used in the algorithm could steer the research AGV in the direction of a waypoint. This test was performed outside with good visibility to the GPS satellites. The test waypoint was selected twelve meters from the research AGV The research AGV was started pointing away from the waypoint. The algorithm was set to detect the waypoint within five metres of its location. The test results were very successful. Once the research AGV GPS algorithm started, the AGV turned around towards the waypoint. This test was repeated several times from various directions from the waypoint, with the AGV stopping five meters from the waypoint each time.

The second test is to determine how accurate the algorithm can navigate the research AGV to a predefined waypoint. This test was performed outside with good visibility to the GPS satellites. The waypoint was selected twelve meters from the research AGV. The accuracy of the algorithm was set to detect a waypoint within one metre of its location. The test result showed that the research AGV moved in the correct direction, but would not consistently detect the position on the first pass. The AGV would overshoot the position and then turn around once it had passed. The AGV would eventually locate the position. It was concluded that one meter was not achievable and the distance to waypoint variable was adjusted to find the most reliable value. Next the test was adjusted to detect the waypoint position within 3 meters of it location, this setting proved to be the most reliable.

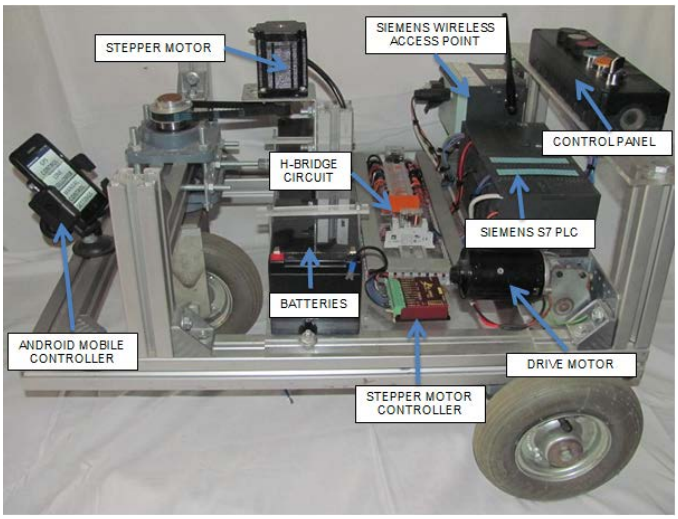

Figure 6: The Complete Research AGV

The purpose of the final test is to determine whether the research AGV can navigate to multiple waypoints. This test was performed outside in a parking lot with good visibility to the GPS satellites. Four waypoints were selected with each of them being approximately twelve metres away from each other. The algorithm was set to detect a waypoint within three metres of each waypoint. The test was performed successfully and the research AGV was able to navigate to all the waypoints.

6 CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

The navigation system of an AGV is of one of the most important parts of modern AGV's. The development of an AGV navigation system requires thorough research and rigorous testing. In this dissertation an understanding of the various AGV technologies was gained through literature study, software development and various tests. The hardware components of the research AGV were selected based on the knowledge gained through the literature study. Software was developed that would interface between the Android device and the PLC that was selected. Two navigation algorithms were selected for use on the Android device: the line following algorithm using the camera of the Android device and the GPS algorithm implementing the GPS and on-board compass of the Android device.

The implemented line following algorithm was proved through testing. The line following algorithm success was limited owing to the size and speed of the research AGV. Testing revealed that the size of the camera and the angle of the Android device dramatically influence results.

The GPS algorithm that was implemented successfully navigated to various waypoints. The GPS on the Android platform was not accurate. The accuracy would be unsatisfactory if implemented in an industrial environment with it only being able to consistently reach waypoints within 3 meters. One possible explanation for the inaccuracy could be due to the accuracy of the GPS device logging the waypoint positions. If the initial logged position is inaccurate the Android device will locate the waypoint position based on its own accuracy. This combined with the Android GPS accuracy variations causes the research AGV to only achieve a three metre accuracy.

6.2 Conclusions

Various recommendations can be made based on the knowledge gained. If a similar three-wheel design is implemented with the drive at the back, it is recommended

that a differential be implemented to assist with cornering of the vehicle. To improve the steering speed on the existing platform, the pulley ratio can be adjusted to increase the size of each step of the stepper motor. The line following algorithm could be changed to look for multiple dark positions to detect the line on the screen and not a single dot. If this is achieved, the algorithm should be able to detect whether no line is present and steer towards the last line that was drawn.

REFERENCES

[1] Dudec, G. J. (2000). Computational Principles of Mobile Robotics. Cambridge

[2] Ranko, Z., & Guzmán, R. (2003, July 21). An auto-guided vehicle controlled by PC. International Conference on Engineering Education, 1.

[3] Pancham, A. (2008, July). A variable sensor system for guidance and navigation of an AGV. School of Mechanical Engineering. Durban: Univeristy of Kwa-zulu Natal.

[4] Kelley, D. (2000, November). Line Detection Using a Digital Camera Encoder:The Newsletter of the Seatle Robotics Society.

[5] Barrette, R. (2012, January 27). Success! Retrieved November 21, 2012, from RickBarrette.org: http://rickbarrette.org/archives/150.

[6] Beam Robotics Wikipedia. (2011, June 2). Steering_Techniques. Retrieved September 2, 2012, from BEAM-Wiki: http://www.beamwiki.org/wiki/Steering_Techniques.

[7] Bell, J. (2008). Basic GPS Navigation: A practical guide to GPS navigation.*Examples:*

[8] Android. (2010, May 19). About the Android Open Source Project. Retrieved March 8, 2012, from Open Source Project:http://source.android.com/about/index.html

[9] European Robotics Research Network. (2008, Februarie 18). Service Robots1.World Robotics 2006, pp. 379-390.

[10] Hoppen, P., Knieriemen, T., & Von Puttkamer, E. (1990). Laser-radar based mapping and navigation for an autonomous mobile robot. Proceedings of the IEEE International Conference on Robotics and Automation. Vol 2, (pp. 948-953).

[11] McComb, G. (2011). Robot Builder's Bonanza, 4th Edition - Application Notes & Bonus Projects .

[12] Melore, P. (2005). What is a PLC? Retrieved September 12, 2012, from PLCS.net:http://www.plcs.net/chapters/whatis1.html

[13] Mroszczyk, J. W. (2010). Safety Practices For Automated Guided Vehicles. Retrieved September 8, 2012, from The American Society of Safety Engineers: http://www.asse.org/ practicespecialties/ tech/tech2.php

[14] Murphy, M. L. (2008). The Busy Coder's Guide to Android Development. CommonsWare.

[15] Paul, A. (2005). Design Of An Autonomous Navigation System For A Mobile Robot. Department of Bioresource Engineering. Montreal, United States of America: McGill University.